REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(LEAVE BLANK)*

2. REPORT DATE

1998

3. REPORT TYPE AND DATES COVERED

Professional Paper

4. TITLE AND SUBTITLE

Robust Command Augmentation System Design Using Genetic Methods

5. FUNDING NUMBERS

6. AUTHOR(S)

G.D. Sweriduk, P.K. Menon, and M.L. Stienberg

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Air Warfare Center Aircraft Division
22347 Cedar Point Road, Unit #6
Patuxent River, Maryland 20670-1161

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

ROBUST COMMAND AUGMENTATION SYSTEM DESIGN USING GENETIC METHODS

G. D. Sweriduk* and P. K. Menon†
Optimal Synthesis, Inc.
450 San Antonio Road, Suite 46
Palo Alto, CA 94306

M. L. Steinberg‡
Aeromechanics Division
Naval Air Systems Command
Patuxent River, MD 20670

Abstract

This paper describes the use of a genetic search method in the design of a command augmentation system for a high-performance aircraft. A genetic algorithm is used in the design of H_∞ controllers for the longitudinal and lateral-directional channels by selecting the weighting functions. The integral of absolute value of error between the actual response and that of an ideal model is used as the fitness criterion, along with additional terms to penalize for cross-coupling between p_S and n_y, non-minimum phase behavior, and the closed-loop infinity-norm bound, γ. Starting from an initial population of weighting functions, the algorithm generates new functions with the goal of improving the fitness. These controllers are then evaluated in a 6 degree-of-freedom nonlinear model of the aircraft.

I. Introduction

The most common approach to flight control law design is gain-scheduling, which requires the design of control laws for a large matrix of flight conditions. Each design can be a time-consuming process, and there is generally a significant amount of trial-and-error involved. Most control law designs have favored classical techniques, where there are a large number of choices in structure and parameters that need to be made by the designer. There is also currently great interest in using multivariable approaches to improve the design process. However, it can be challenging to relate parameters like weighting matrices in multivariable control approaches to the complex quantitative and qualitative design requirements of a flight control law.[1] The purpose of this paper is to present preliminary results of an ongoing research effort into the use of genetic search methods to aid in controller design. The goal of this work is to help automate and accelerate the flight control system design process. Genetic methods are seen as potentially useful to automate certain of the more trial-and-error parts of the flight control design process.

Genetic methods have several advantages as an optimization technique. First, a wide variety of different types of fitness criteria (cost functions) may be used for optimization. This includes the use of discontinuous or non-smooth functions. As a result, the designer has a great deal of freedom, as well as the ability to choose criteria that more closely represent the actual design goals, rather than having to adapt these goals to meet the needs of a more restrictive optimization approach. Another advantage of a genetic search is that it can be used to directly design control *laws*, rather than just to find control law parameters. The structure of the controller does not need to be specified in advance, with only some numerical constants to be optimized. Properly set up, a genetic search can piece together mathematical functions to form control laws[2]. On the other hand, the primary disadvantage of using a genetic optimization approach is there is no guarantee that the controller chosen is optimal or near-optimal in any sense with regards to the chosen cost criteria. Also, genetic optimization takes considerable computer time, and this may make it impractical for design changes that require fast turn-around time, like working with a pilot to improve handling qualities in a simulator.

In this paper, a genetic search method is used to design H_∞ controllers for a high-performance aircraft by selecting the weighting functions. Fitness is determined by comparing the closed-loop response with a Level 1 flying qualities model. Figure 1

---

* Research Scientist, Senior Member, AIAA
† President, Associate Fellow, AIAA
‡ Flight Control Engineer, Senior Member, AIAA

illustrates the overall design process. An initial population of weighting parameters is chosen to start the process. Next, new members are synthesized using genetic operations. These new members are then used to synthesize H_∞ controllers that are evaluated through simulations. Finally, these new members are added to the population.

Section II discusses genetic search methods, Section III contains the problem setup, and Section IV presents results.

II. Genetic Algorithms and Genetic Programming

Only a brief description of genetic algorithms and genetic programming will be given in this paper. The interested reader is referred to the references herein for more information.

The basic concept of genetic algorithms was introduced by Holland[3], who showed how the evolutionary process could be applied to artificial systems[4]. The genetic algorithm is a mathematical algorithm which transforms a set (population) of mathematical objects (members) into a new set using operations similar to the process of natural selection, as described by Charles Darwin in his well-known treatise[5]. The main operations are reproduction, crossover, and mutation. Each new population is called a generation. The fitness of each member of the current generation is evaluated according to some specified function. The members with the best fitness are more likely to be selected to be carried over to the next generation (reproduction) or used to create offspring (crossover) which will be included in the next generation. Members with poor fitness are more likely to be eliminated from the population. Members can also be selected at random and altered (mutation).

Throughout the 1980's, extensions to the standard genetic algorithm were proposed[3,6]. In the standard genetic algorithm the members are usually fixed-length strings. The strings are made up of binary numbers which can represent real numbers or just actions (i.e. fast vs. slow, high vs. low, etc.). A string therefore represents a set of numbers or a sequence of actions. By breeding and mutating these strings, new combinations are formed, and the new strings are evaluated for fitness. However, the length of the string, and the sturcture of the solution, is always fixed in genetic algorithms. In the genetic programming methodology,[3] the complexity of the members undergoing adaptation is much greater. The members may be rules such as logical operators. In this way a genetic algorithm can be used to create computer programs to solve a specific problem. According to Ref. 3, "...the structures undergoing adaptation in genetic programming are active. They are not passive encodings of the solution to the problem. Instead, given a computer on which to run, the structures in genetic programming are active structures that are capable of being executed in their current form."[3]. It is this process of creating programs that leads to the term "genetic programming." There are variations of the classical genetic algorithm and genetic programming, but they all share the basic concepts, so they are referred to collectively as genetic search methods.

In Ref. 2, genetic search methods were used to design nonlinear control laws for a model of an A-4 aircraft. Both autopilot controllers and guidance laws were developed using this methodology. Other The present study will use a genetic search to design a command augmentation system (CAS) with H_∞ control using linearized models.

III. Command Augmentation System (CAS) Synthesis

Although the genetic search methodology can handle models of arbitrary complexity, in the interests of saving computer time, linearized aircraft models are used as the basis for CAS design. Linear models were obtained from a nonlinear, 6 degree-of-freedom model. From the full-order linearized model, a second-order longitudinal model and a fourth-order lateral-directional model were extracted. The state variables of the longitudinal model are q (pitch rate) and w (body z-axis velocity), the feedback variables are q and n_z (normal acceleration), and the input is the stabilator. Normal acceleration will be the commanded quantity. The state variables of the lateral-directional model are v (body y-axis velocity), ϕ (roll angle), p (body axis roll rate), and r (body axis yaw rate). The feedbacks are p_s (stability axis roll rate), r, and n_y (lateral acceleration), and the inputs are effective aileron and rudder. Roll control is achieved with a combination of ailerons and differential stabilator, which is referred to here as effective aileron. The quantities to be commanded in the lateral channel are p_s and n_y.

Two separate controllers will be designed, one for the longitudinal dynamics and one for the lateral-directional dynamics. Figures 2 and 3 show the generalized plants which are to be used for the designs. There are poles, zeros, and gains of the weighting functions, which the engineer would select based on certain guidelines and also trial-and-error.

The weights will be expressed in terms of free parameters. The performance variables are the weighted control signals and the errors between actual outputs and the outputs of ideal models.

The ideal models represent dynamics that are classified as Level 1 handling qualities[7]. The ideal model for the short-period mode is the second-order transfer function:

$$f_{sp}(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

The values of ζ and ω are chosen to be 0.7 and 3, respecively. The ideal model for the roll mode is:

$$f_r(s) = \frac{a}{s + a}$$

where a is chosen to be 2.5. The ideal model for the Dutch roll mode is also a second-order transfer function like $f_{sp}(s)$, and the values of ζ and ω are chosen to be 0.7 and 1.5, respectively. Uncertainties are included at the plant input in the longitudinal case and at the output in the lateral-directional case, and disturbances are included in all measurements.

The weights are chosen by the genetic algorithm and the controller is designed with commercially available H_∞ design software (MATLAB® with μ-Tools® Toolbox). In the longitudinal case, the free parameters determine the weights as follows:

$$W_{s1} = \frac{K5(s + (K6 + K7))}{(s + K7)}, \quad W_{s2} = K4$$

$$W_\Delta = \frac{1/(1 + K1)(s + (K2 + K3))}{(s + K3)}$$

$$W_u = K8$$

$$W_{d1} = W_{d2} = 0.01$$

where K1 - K8 are the free parameters and represent real numbers. Both weights W_{s1} and W_{s2} are on error signals between the actual output and the output of an ideal model. The penalty is higher at low frequencies and lower at high frequencies, so the zero should be faster than the pole. Since the emphasis is on normal acceleration, less weight is placed on pitch rate, and hence W_{s2} is expected to be a small, constant value. For W_Δ, the uncertainty is expected at low frequency, and low gain above that, so the zero is faster than the pole, and the gain is constructed to ensure the gain is less than 1 at high frequencies. W_u is the weight on control and is a constant value over all frequencies. A frequency-dependent weight could have been used for the control variables also, but this approach seems to work and keeps the order of the controller, and the number of free parameters, smaller. W_{d1} and W_{d2} are noise levels and are held fixed. Typical frequency responses for W_{s1} and W_Δ are shown in Fig. 4.

The free parameters in the lateral-directional case K9 - K17 determine the weights as follows:

$$W_{s1} = \frac{1/(1+10*K9)\left(s+10^{-4}\right)\left(s+(K10+K11+K12)\right)}{\left(s+10^{-4}+K11\right)\left(s+10^{-4}+K12\right)}$$

$$W_{s2} = \frac{1/(1+10*K13)(s+K14+K15)}{s+K15}$$

$$W_{t1} = W_{t2} = W_{t3} = \frac{0.01(s+1)}{s+10}$$

$$W_{u1} = K16, \quad W_{u2} = K17$$

$$W_{d1} = W_{d2} = W_{d3} = 10^{-3}$$

Both weights W_{s1} and W_{s2} are on error signals between the actual output and the output of ideal models. The weight for roll rate, W_{s1}, is largest in the middle frequencies. A zero at very low frequency (10^{-4} rad/s) is inserted to reduce gain at low frequency. There are two poles faster than this zero, and another zero above the poles. The gain is constructed to ensure that the magnitude is less than 1 at high frequencies, and the factor of 10 was included increase the effectiveness of K9 and K13 regardless of their values. The weight on lateral acceleration, W_{s2}, is constructed to have high gain at low frequency and gain less than one at high frequency. The weights on output uncertainty, W_{t1}, W_{t2}, and W_{t3}, are fixed for all designs and reflect increased uncertainty at higher frequencies. The weights on controls, W_{u1} and W_{u2}, are constants. The weights on noise, W_{d1}, W_{d2}, and W_{d3}, are held fixed at a small value.

The frequency-dependent weights were kept as simple as possible in order to keep the controller order low, thereby circumventing the need for order reduction. While the structure of the weights is fixed in this work, it should be possible to allow the structure to vary, although this would create more

complexity in the design problem, with attendant difficulties.

The values of the free parameters for the weights have the following parametrization:

$$u_i = 1,..., 9, 10^0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^1, 10^2$$
$$i = 1,..., 16$$

$$K(n) = u_i * u_j$$

Each of the free parameters K(n) will have its own separate population. A member, then, for this work, means a set of members, each drawn from one of the K(n) populations. This set will be evaluated and assigned a value of fitness, so that each element of the set has the same fitness associated with it.

A set of parameters is sought for designing the H_∞ controllers. While the problem could be set up using binary strings, as in the classical genetic algorithm, an alphanumeric parametrization is used instead, which adds flexibility. The standard genetic algorithm represents choices with binary strings of fixed length. Such an arrangement describes discrete values. Using the above parametrization, and assuming fixed strings with only one multiplication, there would be 16 x 16 = 256 possible values for each free parameter. For H_∞ design, the poles, zeros, and gains of the weighting functions are seldom required to have more than one or two significant digits, so the expressions can be kept fairly simple. Any positive, real number could be represented by multiplication and addition of the parameters u_i if expressions were allowed to grow longer with more operations. However, although expressions are manipulated, the end result is still just the formation of real numbers without affecting the structure of the control law. Hence, the present work really employs a genetic algorithm without using binary strings.

In order to motivate the application of genetic search methods for the present CAS design problem, plots (two views) of the fitness with respect to some of the parameters in the weighting functions are given in Figs. 5 and 6. These figures are for simultaneous variations in K3 and K8. Not all values of K produce a controller, so for those values which did not, the fitness was set to 10^6. From these figures, it may be observed that some conventional search techniques are likely to fail due to the discontinuous and mostly non-smooth behavior of the search task. There are also several local minima. There is no guarantee that genetic methods will find a global minimum, although the chances increase the more generations that are run. While this is not a rigorous examination, it gives some indication that genetic methods may have distinct advantages over other optimization methods for solving this problem.

A member is evaluated by substituting the values of the free parameters into the block diagrams, forming the generalized plant, obtaining the H_∞ controllers, and then simulating the closed-loop system. Not all members will produce a controller. The controller synthesis function requires an initial guess for the upper and lower bounds of the closed-loop infinity norm. The upper limit is set fairly high, and any member which does not saitisfy the upper bound is rejected. Given a controller, fitness is determined by evaluating the responses of the closed-loop system with various inputs. The inputs are step functions to normal acceleration in the longitudinal case and roll rate and lateral acceleration in the lateral-directional case. An example of how the fitness is determined is shown in Fig. 7. The idea is to try to match the behavior of the ideal model, or minimize the error between the ideal model and the actual system. The goal, then, is to obtain the lowest possible fitness. Additional terms are computed to penalize cross-coupling between p_s and n_y, non-minimum phase behavior, and the closed-loop infinity-norm bound, γ, and added to the fitness. Also, for practical reasons, controllers with right-half-plane poles or very fast poles are rejected.

The initial population is generated randomly. After evaluation of the initial population, the process of crossover is carrried out to generate new members. A maximum population size is maintained, and members with high fitness are deleted from the populations. This process is continued until a desirable response is achieved. In evolutionary processes, it is difficult to say that any particular member is optimal, so deciding when to stop the genetic algorithm is somewhat subjective. In theory, running the algorithm longer will continue to produce better results. Several runs are usually conducted.

The genetic search method used in this work differs from the classical algorithm in a few ways. Firstly, only one type of operation, crossover, is used. Secondly, selection of the members for genetic operations is done randomly rather than on the basis of fitness. Thirdly, in the traditional meaning of genetic methods, a generation is the population after all members have been acted upon. However, in this work, a generation refers to the population after one pair of members has undergone crossover. In the traditional sense, all of the members have undergone

reproduction (replication) except the two which have undergone crossover.

IV. Results

Designs have been completed for several flight conditions, and results using a nonlinear aircraft simulation for the flight condition 19000 ft., Mach 0.8 will be presented here. For the longitudinal case, four runs were made, each starting with a population of 500 members, a maximum population size of 500, and lasting 1000 generations. The fourth run produced the best result. Typically the best fitness of the initial population was around 0.02. The best fitness after 1000 generations was 0.00988. While the changes may appear small, the differences are significant. The controller with the lowest fitness produced a closed-loop infinty norm of 0.9780, and the simulation showed good performance. In the lateral case, three runs of 500 generations were made (less generations were used in the lateral case due to the slower run time). The second run produced the best result, with a fitness of 0.01979 and a closed-loop infinity norm of 2.4414, and good performance in the simulation.

The closed-loop responses to various inputs are shown in Figs. 8, 9, and 10. The first is the response to a 0.1 g pulse command to normal acceleration, the second response is with a 0.1 rad/sec doublet command to stability axis roll rate, and the third is a 0.1 g pulse to lateral acceleration. The closed-loop system responds well to all three inputs. There is some unavoidable nonminimum phase behavior.

V. Conclusions

With the parametrization used in this study, the genetic algorithm produced good results. The fitness criterion is fairly simple, but it should be possible to add further refinements in the design problem. Fitness could be defined by directly computing handling qualities metrics, similar to the NASA/Army code CONDUIT[8]. Also, the use of nonlinear simulations with the complete aircraft model during optimization would give a more accurate evaluation and thus might be preferable. However, this was impractical for the present study with the available computer resources, given the computation time required for nonlinear simulations. In addition, more study of the effects of genetic search parameters such as initial and maximum population size is needed.

VI. References

1 Enns, D., et al., "Application of Multivariable Control Theory to Aircraft Control Laws; Interim Report: Definition of Control Problems and Requirements," Air Force Report WL-TR-96-3071, January, 1996.

2 Menon, P.K., Yousefpor, M., Lam, T., and Steinberg, M.L., "Nonlinear Flight Control System Synthesis Using Genetic Programming," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, New Orleans, LA, 1996,

3 Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

4 Darwin, C.A., *On the Origin of Species by Means of Natural Selection*, John Murray, London, 1859.

5 DeJong, K.A., "On Using Genetic Algorithms to Search Program Spaces," *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, J. J. Grefenstette, editor, Erlbaum Press, 1987.

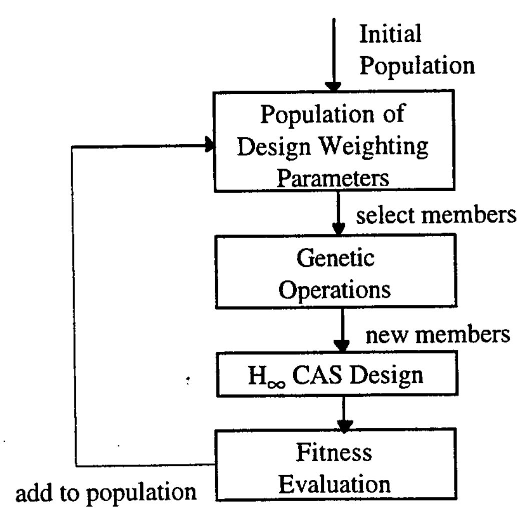
6 Koza, J.R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, 1992.

7 United States Department of Defense, "Flying Qualities of Piloted Vehicles, MIL-STD-1797", March, 1987.

8 Tischler, M.B., et al., "CONDUIT - A New Multidisciplinary Integration Environment for Flight Control Development," NASA Technical Memorandum 112203, June 1997.

Acknowledgements

This work is being supported by the United States Navy, contract number N62269-5361-1291. The authors would like to acknowledge the contribution of the Boeing Company to this work, especially by Kevin Wise, Joe Brinker, Kevin Saari, and Jeff Zimmerman.



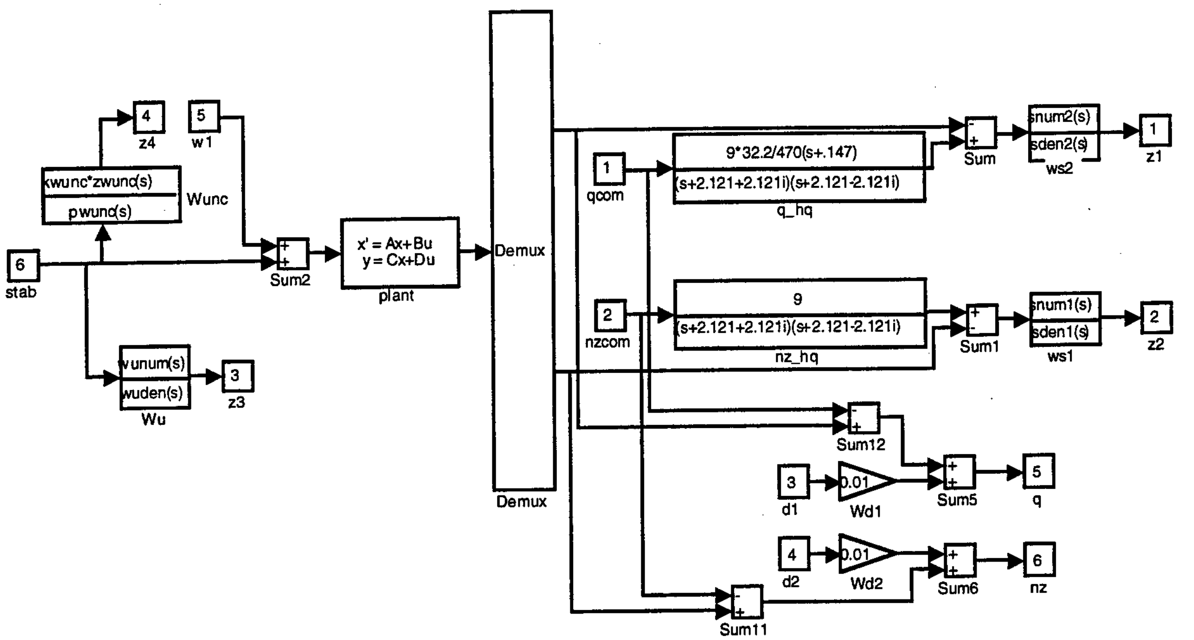


Figure 2: Longitudinal Block Diagram

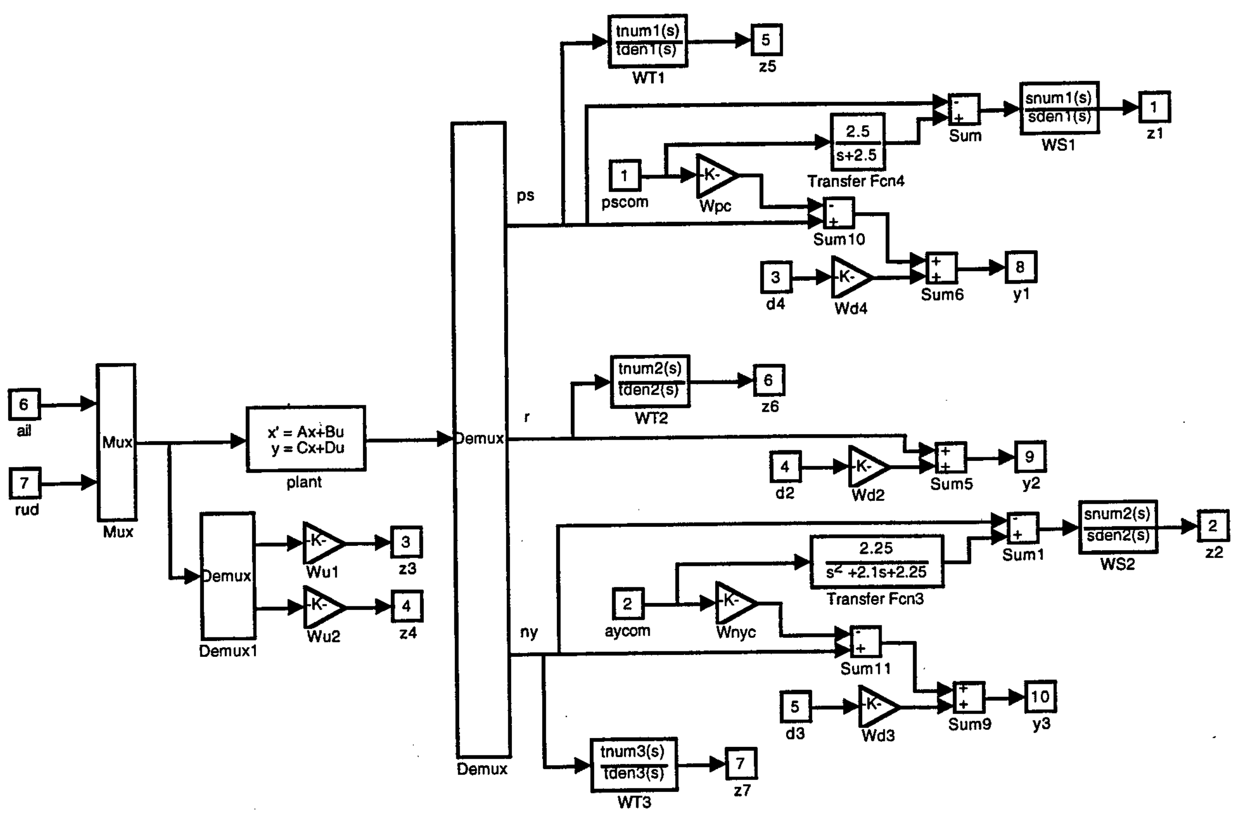


Figure 3: Lateral-Directional Block Diagram

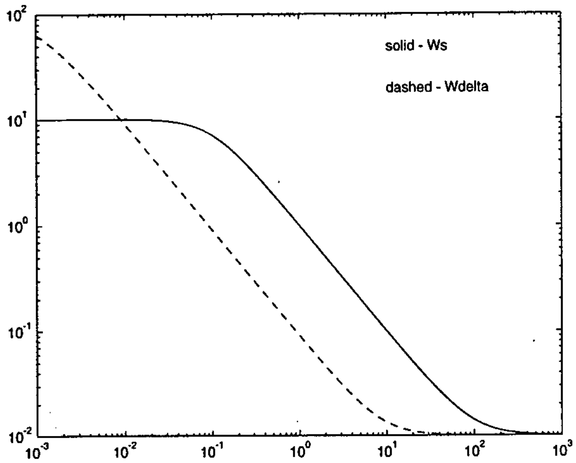


Figure 4: Weighting Function Frequency Response

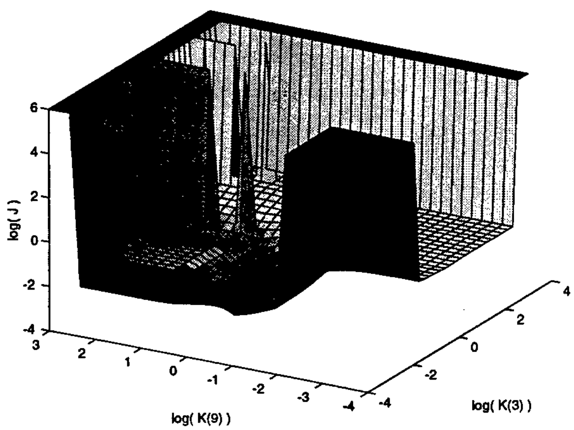


Figure 5: Fitness vs. Parameter Values

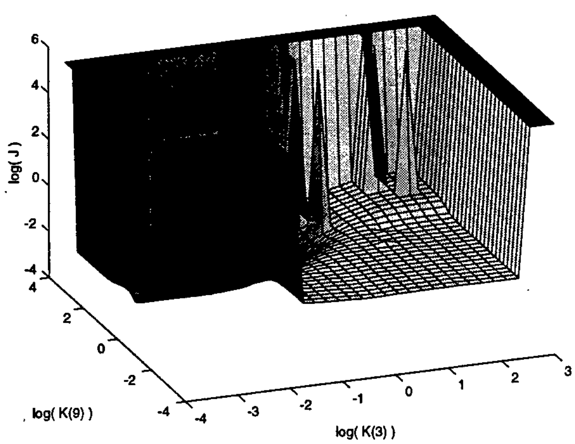


Figure 6: Fitness vs. Parameter Values

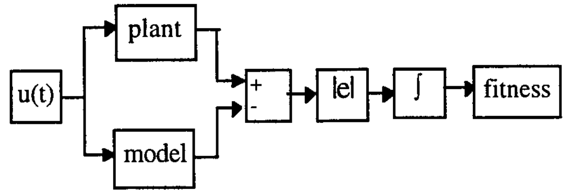


Figure 7: Fitness Calculation

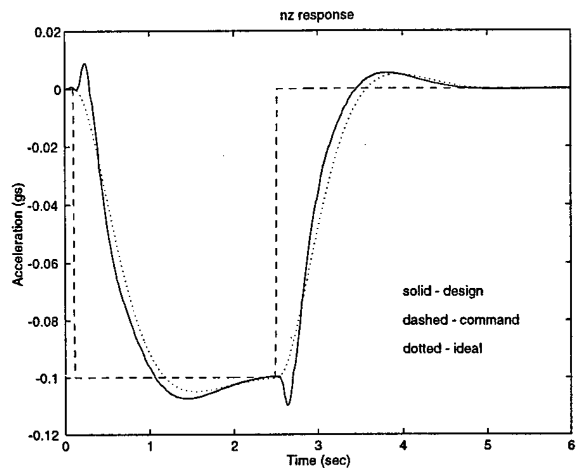


Figure 8: nz response

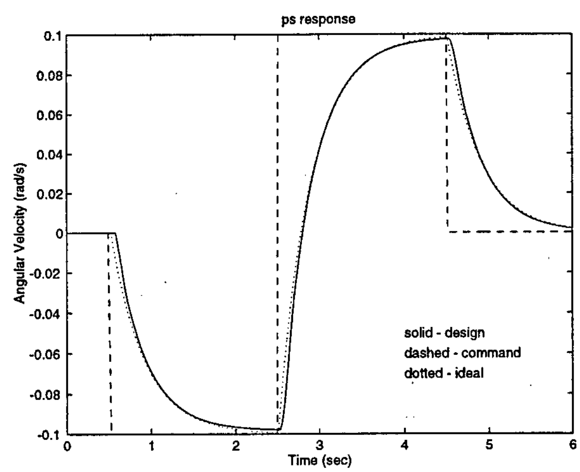


Figure 9: ps response

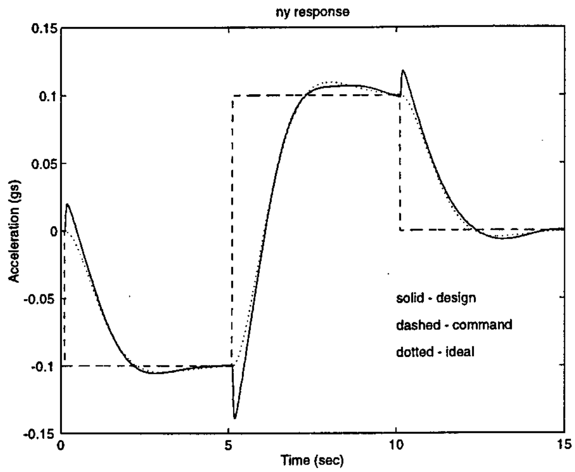


Figure 10: ny response